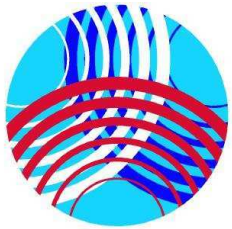
From Causal to z-Linearizable Transactional Memory

Torvald Riegel (Dresden University of Technology, Germany)
Heiko Sturzrehm (University of Neuchatel, Switzerland)
Pascal Felber (University of Neuchatel, Switzerland)
Christof Fetzer (Dresden University of Technology, Germany)



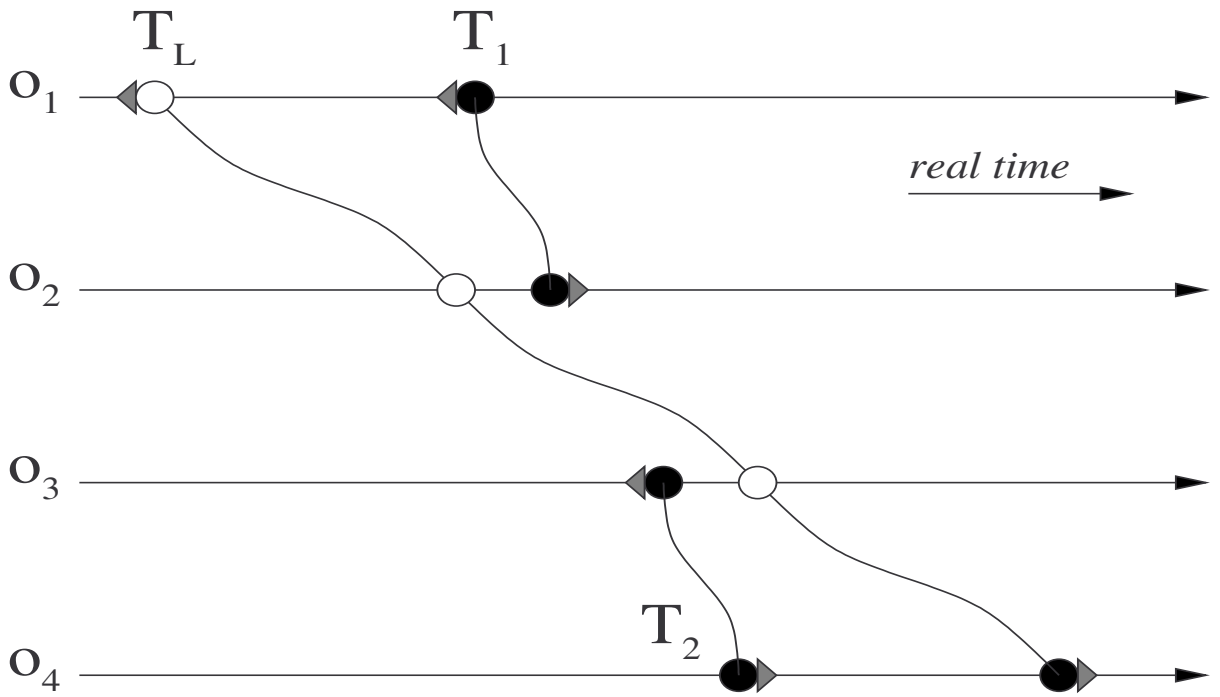
MICS

Improve performance of long transactions

- Long read-only txns: No problem if time-based transactional memory + multiple object versions (see DISC 2006, SPAA 2007)
- Problem: Long/large update transactions:
  - Very low performance in current transactional memories
- What we did:
  - Tried vector clocks for STM in practice
  - Created a faster concurrency control algorithm that provides z-Linearizability

Long update transactions: Serializability vs. Linearizability

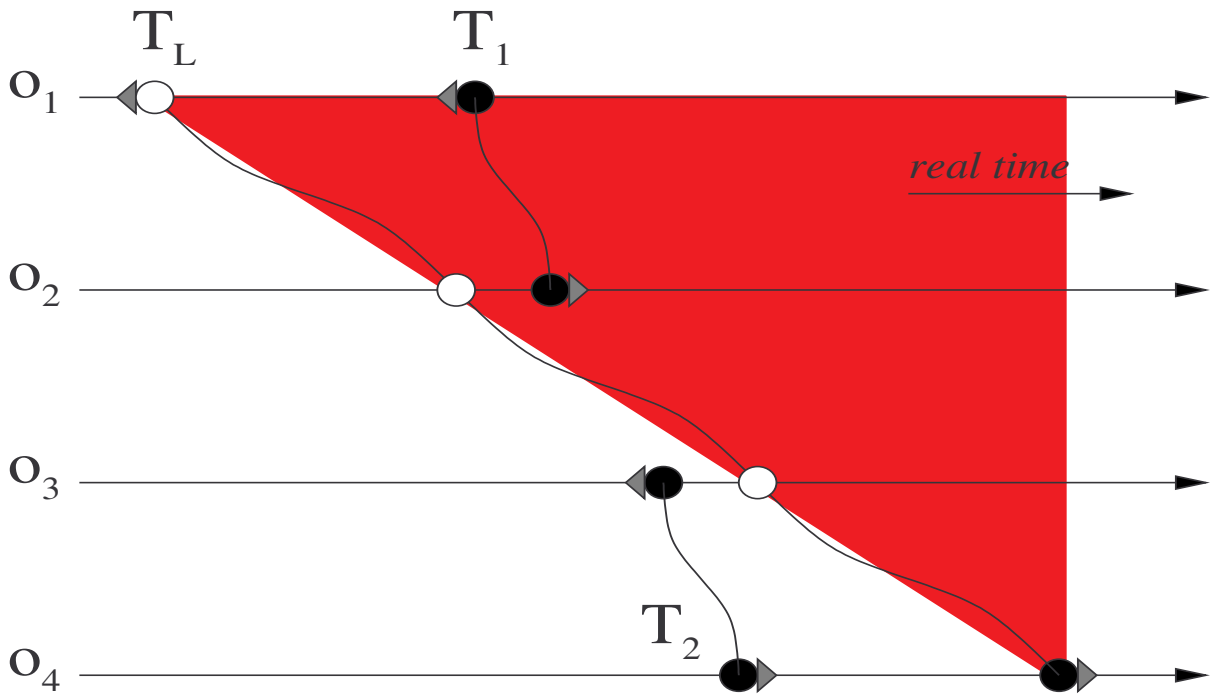
- Current STMs:
  - 2PL or first-committer-wins
  - Gives linearizability
  - Reduces level of concurrency for long update transactions



- Serializability is not evil!
- If threads only run txns via STM (program analysis), they can't detect difference from linearizability

Long update transactions: Serializability vs. Linearizability

- Current STMs:
  - 2PL or first-committer-wins
  - Gives linearizability
  - Reduces level of concurrency for long update transactions

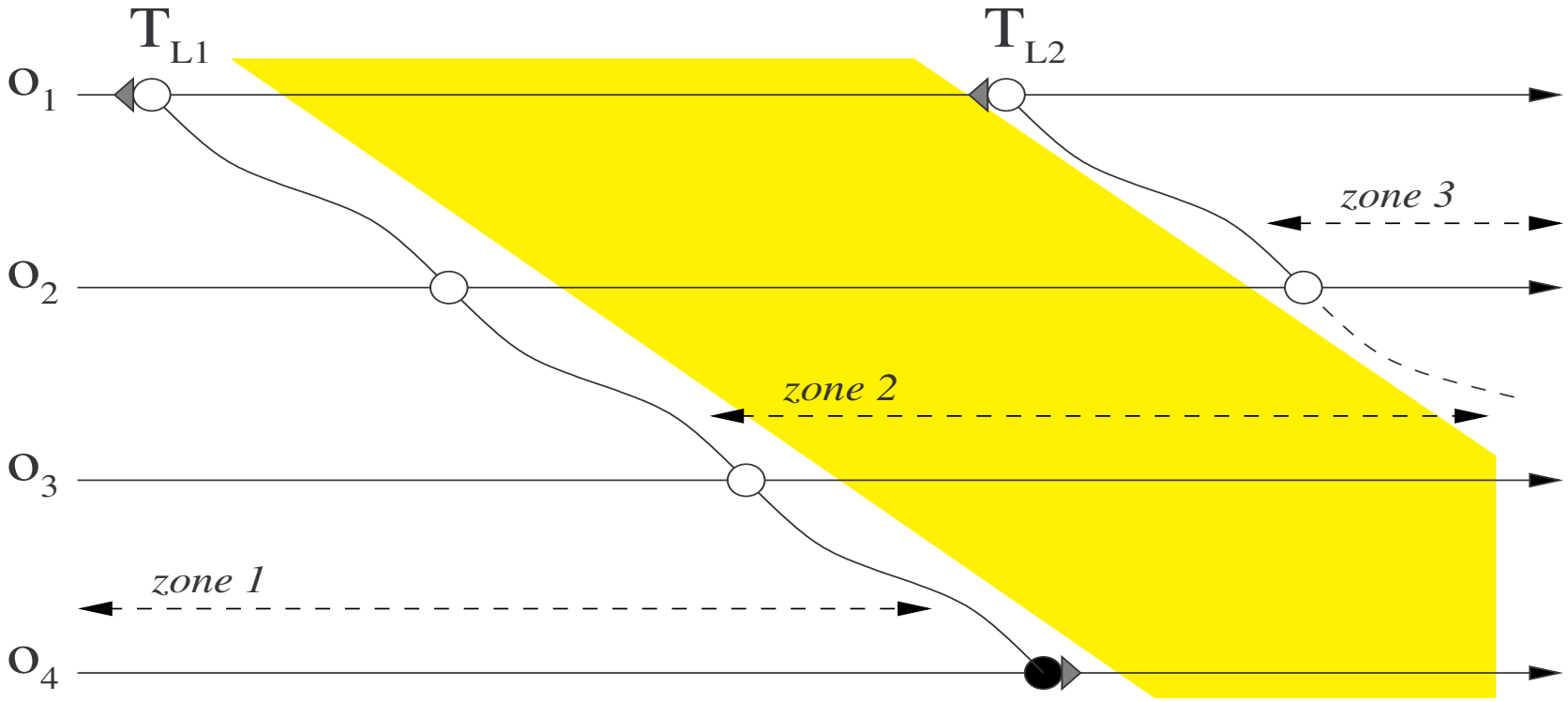


- Serializability is not evil!
- If threads only run txns via STM (program analysis), they can't detect difference from linearizability

Vector clocks as time base for transactional memory

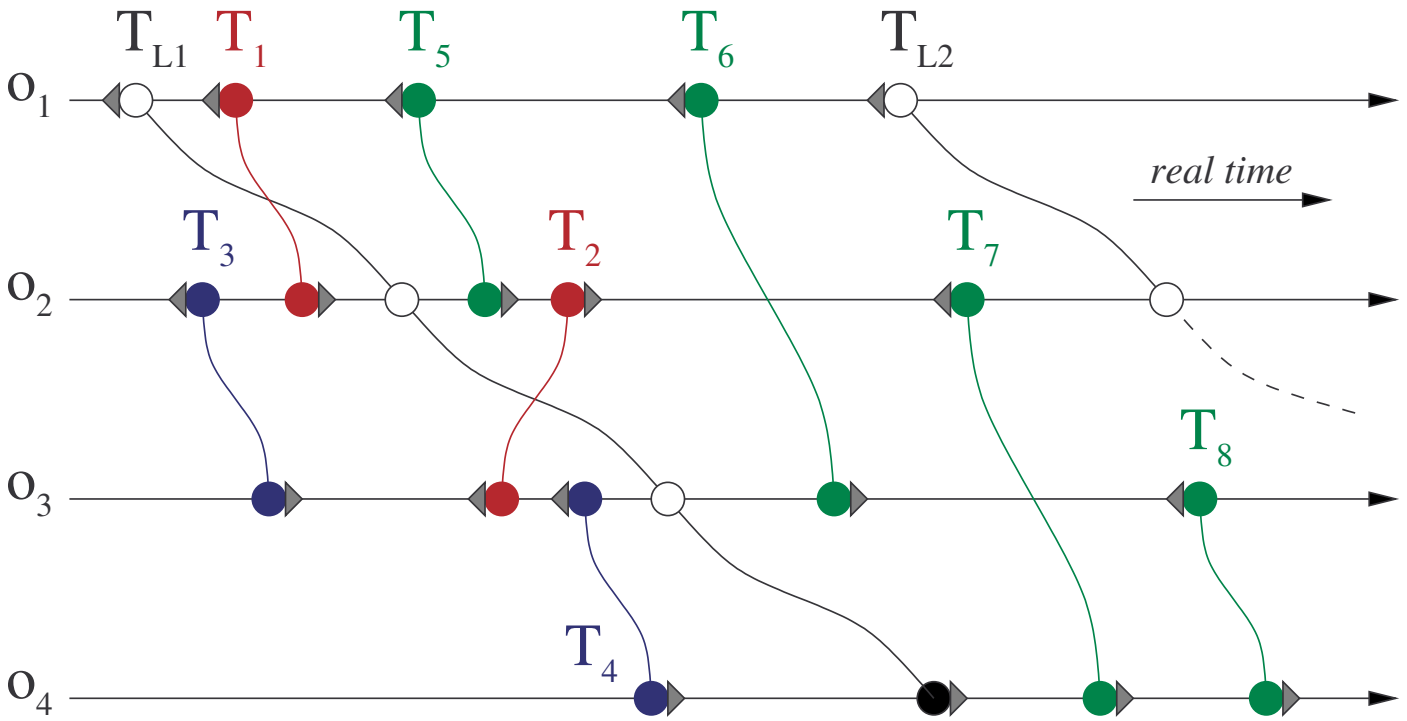
- Vector clocks with one element per processor, or plausible clocks
- Object versions timestamped with txn commit time
- Causal Serializability:
  - Check requires moderate overhead
  - But weaker than serializability
- Serializability:
  - Expensive construction of partial precedence graphs for causally preceding active txns
- Disadvantage: High runtime overhead

z-Linearizability



- Idea: Large txns partition other txns into time zones
- Zone borders are stored in object versions
- Zones are created by accesses by large txns

z-Linearizability



- Red txns access objects in different zones ("cross" first long txn T_{L1})
- Red txns will be aborted (at time of bad access)
- Short txns don't need to wait for long txns (e.g., T_5)

z-Linearizability: Results

- Guarantees serializability for all txns, linearizability in each timezone (can switch to linearizability for all txns when necessary)
- Can combine existing algorithms (e.g., time-based) with special algorithm for long txns
- Short txns: only 1 additional comparison per access
- Only 1 additional timestamp per object version
- Long txns: Set and check zone only, no unlock phase on commit
- Performance results look good, per-access overheads for short txns are moderate